Theoretical Guarantees for Learning Weighted Automata

Borja Balle

Mathematics & Statistics | Lancaster University

ICGI Keynote — October 2016

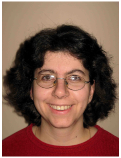
Thanks To My Collaborators!

Mehryar Mohri

Ariadna Quattoni

Xavier Carreras

Prakash Panangaden



Doina Precup

Outline

1. The Complexity of PAC Learning Regular Languages
2. Empirical Risk Minimization for Regular Languages
3. Statistical Learning of Weighted Automata via Hankel Matrices

Outline

Regular Inference (Informal Description)

- Unknown regular language $L \subseteq \Sigma^\star$
  - With indicator function $f : \Sigma^\star \to \{0, 1\}$
- Given examples $(x^1, f(x^1)), (x^2, f(x^2)), \ldots$
  - Finite or infinite
  - (positive and negative) OR (only positive)
- Find a representation for L (eg. a DFA)
  - Using a reasonable amount of computation
  - After seeing a reasonable amount of examples

PAC Learning Regular Languages

- Concept class $\mathcal{C}$ of functions $\Sigma^\star \to \{0,1\}$
  - Eg. $\mathcal{C} = \mathcal{DFA}_n$ all regular languages recognized by DFA with n states
- Hypothesis class $\mathcal{H}$ of representations for functions $\Sigma^\star \to \{0,1\}$
  - Proper learning $\mathcal{H} = \mathcal{C}$
  - Improper learning $\mathcal{H} \neq \mathcal{C}$

Definition: PAC Learner

An algorithm $\mathtt{A}$ such that for *any* $f \in \mathcal{C}$ and *any* prob. dist. D on $\Sigma^\star$, and *any* accuracy ε and confidence δ, satisfies: *given a large enough sample of examples* $S = ((x^i, f(x^i)))$ *i.i.d. from* D*, the output hypothesis* $\hat{f} = \mathtt{A}(S) \in \mathcal{H}$ *satisfies* $\mathbb{P}_{x\sim D}[f(x) \neq \hat{f}(x)] \leqslant \varepsilon$ *with probability at least* $1-\delta$.

- *Large enough* typically means polynomial of $1/\varepsilon$, $1/\delta$, size of f
- For any prob. dist. D on $\Sigma^\star$ is called *distribution-free learning*

Note: see [De la Higuera, 2010] for other important formal learning models

Sample Complexity of PAC Learning DFA

Sample Complexity

The distribution-free sample complexity of PAC learning $\mathcal{C} = \mathcal{DFA}_n$ is polynomial in n and $|\Sigma|$

Follows from:

- Any concept class $\mathcal{C}$ can be proper PAC-learned with:
  - $|S| = O\left(\frac{\mathsf{VC}(\mathcal{C})\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\right)$ [Vapnik, 1982]
  - $|S| = O\left(\frac{\mathsf{VC}(\mathcal{C})\log(1/\delta)}{\varepsilon}\right)$ [Haussler et al., 1994]
  - $|S| = O\left(\frac{\mathsf{VC}(\mathcal{C})+\log(1/\delta)}{\varepsilon}\right)$ [Hanneke, 2016]
- $\mathsf{VC}(\mathcal{DFA}_n) = O(|\Sigma| n \log n)$ [Ishigami and Tani, 1993]

Generic Learning Algorithm:

- Upper bounds in [Vapnik, 1982, Hanneke, 2016] apply to *consistent* learning algorithms
- A is *consistent* if for any sample $S = ((x^i, f(x^i)))$ the hypothesis $\hat{f} = \mathtt{A}(S)$ satisfies $\hat{f}(x^i) = f(x^i)$ for all i

Computational Complexity of PAC Learning DFA

- Proper PAC learning of DFA is equivalent to finding *smallest consistent DFA* with S [Board and Pitt, 1992]
- Finding the smallest consistent DFA is NP-hard [Angluin, 1978, Gold, 1978]
- Approximating the smallest consistent DFA is NP-hard [Pitt and Warmuth, 1993, Chalermsook et al., 2014]
- Improper learning DFA is as hard as breaking RSA [Kearns and Valiant, 1994]
- Improper learning DFA is as hard as refuting random CSP [Daniely et al., 2014]

Is Worst-case Hardness Too Pessimistic?

Positive Results:

- Given characteristic sample, state-merging can find smallest consistent DFA [Oncina and García, 1992]
- PAC learning is possible under nice distributions adapted to target language [Parekh and Honavar, 2001, Clark and Thollard, 2004]
- Random DFA under uniform distributions seem easy to learn [Lang, 1992, Angluin and Chen, 2015]
- And also lots of successful heuristics in practice: EDSM, SAT solvers, etc.

Take Away:

- By giving up on distribution-free and focusing on *nice distributions* efficient PAC learning is possible
- Almost all of these algorithms still focus on *sample consistency*
- Do we expect them to work well for *practical applications*?
  - Probably yes for software engineering
  - Probably not for NLP, robotics, bioinformatics, ...

Outline

1. The Complexity of PAC Learning Regular Languages
2. Empirical Risk Minimization for Regular Languages
3. Statistical Learning of Weighted Automata via Hankel Matrices

Regular Inference as an Optimization Problem

Thought Experiment

Given input sample $S = ((x^i, y^i))$ for $i = 1...100$, would you rather:

1. classify all 100 examples correctly with 50 states, or
2. classify 95 examples correctly with 5 states?

Optimization Problems

1. Minimal consistent DFA

$$\min_{A \in \mathcal{DFA}} |A| \quad \text{s.t. } A(x^i) = y^i \ \forall i \in [m]$$

2. Empirical risk minimization in $\mathcal{DFA}_n$

$$\min_{A \in \mathcal{DFA}} \frac{1}{m} \sum_{i=1}^{m} 1[A(x^i) \neq y^i] \quad \text{s.t. } |A| \leqslant n$$

Statistical Learning for Classification

Statistical Learning Setup

- D probability distribution over $\Sigma^\star \times \{+1, -1\}$
- $\mathcal{H}$ hypothesis class of functions $\Sigma^\star \to \{+1, -1\}$
- ℓ_{01} the 0-1 loss function for $y, \hat{y} \in \{+1, -1\}$

$$\ell_{01}(\hat{y}, y) = \frac{1 - \mathrm{sign}(\hat{y}y)}{2} = 1[\hat{y} = y]$$

Statistical Learning Goal

- Find the minimizer of the *average loss*:

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_{(x,y)\sim D}\left[\ell_{01}(f(x), y)\right] = \operatorname*{argmin}_{f \in \mathcal{H}} L_D(f; \ell_{01})$$

- From a sample $S = ((x^i, y^i))$ with m i.i.d. examples from D

$$\mathbb{E}_{(x,y)\sim D}\left[\ell_{01}(f(x), y)\right] \approx \frac{1}{m}\sum_{i=1}^{m} \ell_{01}(f(x^i), y^i)$$

ERM and VC Theory

Empirical Risk Minimization (ERM)

- Given the sample $S = ((x^i, y^i))$ return the hypothesis

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell_{01}(f(x^i), y^i,) = \operatorname*{argmin}_{f \in \mathcal{H}} \hat{L}_S(f; \ell_{01})$$

Statistical Justification

- Generalization bound based on VC theory: with prob. at least $1 - \delta$ over S (e.g. see [Mohri et al., 2012])

$$L_D(f; \ell_{01}) \leqslant \hat{L}_S(f; \ell_{01}) + O\left(\sqrt{\frac{VC(\mathcal{H}) \log m + \log(1/\delta)}{m}}\right) \qquad \forall f \in \mathcal{H}$$

- In the case $\mathcal{H} = \mathcal{DFA}_n$:

$$L_D(A; \ell_{01}) \leqslant \hat{L}_S(A; \ell_{01}) + O\left(\sqrt{\frac{|\Sigma| n \log n \log m + \log(1/\delta)}{m}}\right) \qquad \forall |A| \leqslant n$$

Sources of Hardness in ERM for DFA

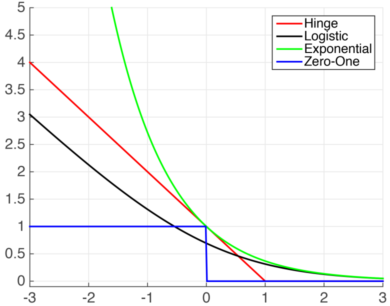
$$\min_{\mathcal{A} \in \mathcal{DFA}} \frac{1}{m} \sum_{i=1}^{m} \ell_{01}(A(x^i), y^i) \quad \text{s.t. } |A| \leqslant n$$

- Non-convex loss: $\ell_{01}(A(x), y)$ is not convex in $A(x)$ because of sign
- Combinatorial search space: search over DFA is search over labelled directed graph with constraints
- Non-convex constraint: introducing $|A|$ into the optimization is hard

Common Wisdom: Optimization tools that work better in practice deal with differentiable and/or convex problems

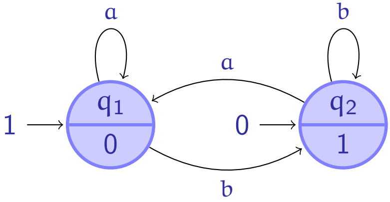
Roadmap to a Tractable Surrogate

- Replace by ℓ_{01} by a convex upped bound



- Make search space continuous: from DFA to WFA
- Identify convex constraints on WFA that can prevent overfitting

Writing DFA with Matrices and Vectors

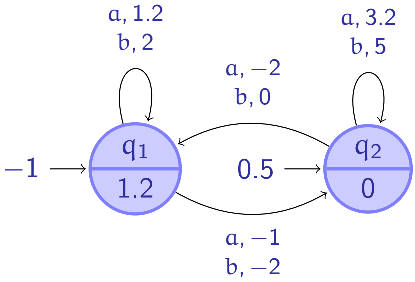


$$\mathcal{A} = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$$

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{A}_a = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \mathbf{A}_b = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\mathcal{A}(aab) = \boldsymbol{\alpha}^\top \mathbf{A}_a \mathbf{A}_a \mathbf{A}_b \boldsymbol{\beta} = 1$$

Weighted Finite Automata (WFA)



$$A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$$

$$\boldsymbol{\alpha} = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} 1.2 \\ 0 \end{bmatrix} \quad \mathbf{A}_a = \begin{bmatrix} 1.2 & -1 \\ -2 & 3.2 \end{bmatrix} \quad \mathbf{A}_b = \begin{bmatrix} 2 & -2 \\ 0 & 5 \end{bmatrix}$$

$$A : \Sigma^\star \to \mathbb{R} \qquad A(x_1 \cdots x_T) = \boldsymbol{\alpha}^\top \mathbf{A}_{x_1} \cdots \mathbf{A}_{x_T} \boldsymbol{\beta}$$

ERM for WFA is Differentiable

$$\min_{A \in \mathcal{WFA}} \frac{1}{m} \sum_{i=1}^{m} \ell(A(x^i), y^i) \quad \text{s.t. } |A| = n$$

with loss $\ell(\hat{y}, y)$ *differentiable on first coordinate*

Gradient Computation

- WFA $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$, $x \in \Sigma^\star$, $y \in \mathbb{R}$, can compute $\nabla_A \ell(A(x), y)$
- Example with $x = abca$ and weights in $\mathbf{A}_a$:

$$\begin{aligned}\nabla_{\mathbf{A}_a} \ell(A(x), y) &= \frac{\partial \ell}{\partial \hat{y}}(A(x), y) \cdot \left(\nabla_{\mathbf{A}_a} \boldsymbol{\alpha}^\top \mathbf{A}_a \mathbf{A}_b \mathbf{A}_c \mathbf{A}_a \boldsymbol{\beta}\right) \\ &= \frac{\partial \ell}{\partial \hat{y}}(A(x), y) \cdot \left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top \mathbf{A}_a^\top \mathbf{A}_c^\top \mathbf{A}_b^\top + \mathbf{A}_c^\top \mathbf{A}_b^\top \mathbf{A}_a^\top \boldsymbol{\alpha}\boldsymbol{\beta}^\top\right)\end{aligned}$$

- Can use gradient descent to "solve" ERM for WFA
- The optimization is highly non-convex, but its commonly done in RNN
- Since $\mathcal{WFA}_n$ is infinite, what is a proper way to prevent overfitting?

Statistical Learning and Rademacher Complexity

- The risk of overfitting can be controlled with generalization bounds of the form: for any D, with prob. $1 - \delta$ over $S \sim D^m$

$$L_D(f; \ell) \leqslant \hat{L}_S(f; \ell) + C(S, \mathcal{H}, \ell) \qquad \forall f \in \mathcal{H}$$

- Rademacher complexity provides bounds for any $\mathcal{H} = \{f : \Sigma^\star \to \mathbb{R}\}$

$$\mathfrak{R}_m(\mathcal{H}) = \mathbb{E}_{S \sim D^m} \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x^i) \right] \quad \text{where} \quad \sigma_i \sim \text{unif}(\{+1, -1\})$$

- For a bounded Lipschitz loss ℓ with probability $1 - \delta$ over $S \sim D^m$ (e.g. see [Mohri et al., 2012])

$$L_D(f; \ell) \leqslant \hat{L}_S(f; \ell) + O\left(\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{m}} \right) \qquad \forall f \in \mathcal{H}$$

Rademacher Complexity of WFA

- Given a pair of Hölder conjugate integers p, q $(1/p + 1/q = 1)$, define a norm on WFA given by

$$\|A\|_{p,q} = \max\left\{\|\boldsymbol{\alpha}\|_p, \|\boldsymbol{\beta}\|_q, \max_{a\in\Sigma}\|\mathbf{A}_a\|_q\right\}$$

- Let $\mathcal{A}_n \subset \mathcal{WFA}_n$ be the class of WFA with n states given by

$$\mathcal{A}_n = \{A \in \mathcal{WFA}_n \mid \|A\|_{p,q} \leqslant 1\}$$

Theorem [Balle and Mohri, 2015b]

The Rademacher complexity of $\mathcal{A}_n$ is bounded by

$$\mathfrak{R}_m(\mathcal{A}_n) = O\left(\frac{L_m}{m} + \sqrt{\frac{n^2|\Sigma|\log(m)}{m}}\right) ,$$

where $L_m = \mathbb{E}_S[\max_i |x^i|]$.

Learning WFA with Gradient Descent

- Solve the following ERM problem with (stochastic) projected gradient descent:

$$\min_{A \in \mathcal{WFA}_n} \frac{1}{m} \sum_{i=1}^{m} \ell(A(x^i), y^i) \quad \text{s.t. } \|A\|_{p,q} \leqslant R$$

- Control overfitting by tuning R (e.g. via cross-validation)
- Can equally solve classification ($y^i \in \{+1, -1\}$) and regression ($y^i \in \mathbb{R}$) with differentiable loss functions
- Risk of *underfitting*: unlikely that we will find the global optimum, might get stuck in local optimum

Outline

1. The Complexity of PAC Learning Regular Languages

2. Empirical Risk Minimization for Regular Languages

3. Statistical Learning of Weighted Automata via Hankel Matrices

Hankel Matrices and Fliess' Theorem

Given $f : \Sigma^\star \to \mathbb{R}$ define its Hankel matrix $\mathbf{H}_f \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$ as

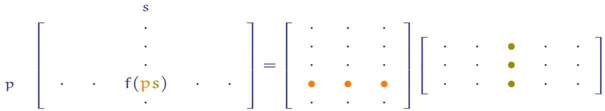
$$
\begin{array}{c} \\ \epsilon \\ a \\ b \\ \vdots \\ p \\ \vdots \end{array}
\begin{array}{c}
\begin{array}{cccccc} \epsilon & a & b & \cdots & s & \cdots \end{array} \\
\begin{bmatrix}
f(\epsilon) & f(a) & f(b) & & \vdots & \\
f(a) & f(aa) & f(ab) & & \vdots & \\
f(b) & f(ba) & f(bb) & & \vdots & \\
 & & & & & \\
\cdots & \cdots & \cdots & & f(ps) & \\
 & & & & &
\end{bmatrix}
\end{array}
$$

Theorem [Fliess, 1974]

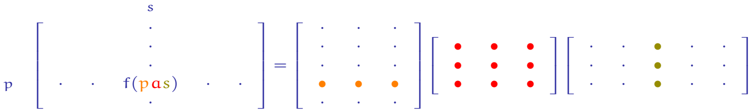
The rank of $\mathbf{H}_f$ is finite if and only if f is computed by a WFA, in which case $\mathrm{rank}(\mathbf{H}_f)$ equals the number of states of a minimal WFA computing f

From Hankel to WFA

$$\mathcal{A}(p_1 \cdots p_T s_1 \cdots s_{T'}) = \boldsymbol{\alpha}^\top \mathbf{A}_{p_1} \cdots \mathbf{A}_{p_T} \mathbf{A}_{s_1} \cdots \mathbf{A}_{s_{T'}} \boldsymbol{\beta}$$



$$\mathcal{A}(p_1 \cdots p_T a s_1 \cdots s_{T'}) = \boldsymbol{\alpha}^\top \mathbf{A}_{p_1} \cdots \mathbf{A}_{p_T} \mathbf{A}_a \mathbf{A}_{s_1} \cdots \mathbf{A}_{s_{T'}} \boldsymbol{\beta}$$



- Algebraically: $\mathbf{H} = \mathbf{P}\mathbf{S}$ and $\mathbf{H}_a = \mathbf{P}\mathbf{A}_a\mathbf{S}$, so we can learn by $\mathbf{A}_a = \mathbf{P}^+\mathbf{H}_a\mathbf{S}^+$
- This is the underlying principle behind query learning and spectral learning for WFA [Balle and Mohri, 2015a]
- For more information, see our EMNLP'14 tutorial with A. Quattoni and X. Carreras [Balle et al., 2014]

Learning with Hankel Matrices [Balle and Mohri, 2012]

Step 1: Learn a finite Hankel matrix over $\mathcal{P} \times \mathcal{S}$ directly from data by solving the convex ERM

$$\hat{\mathbf{H}} = \operatorname*{argmin}_{\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{H}(x^i), y^i) \quad \text{s.t. } \mathbf{H} \in \text{Hankel}$$

Step 2: Extract sub-blocks $\hat{\mathbf{H}}_\epsilon, \hat{\mathbf{H}}_a$ from the Hankel matrix $\hat{\mathbf{H}}$

$$\mathcal{P} \subseteq \mathcal{P}_\epsilon \cup (\mathcal{P}_\epsilon \cdot \Sigma)$$

$$\hat{\mathbf{H}}_\epsilon(p, s) = \hat{\mathbf{H}}(p, s) \qquad p \in \mathcal{P}_\epsilon, s \in \mathcal{S}$$

$$\hat{\mathbf{H}}_a(p, s) = \hat{\mathbf{H}}(pa, s) \qquad p \in \mathcal{P}_\epsilon, s \in \mathcal{S}$$

Step 3: Learn a WFA from the Hankel matrix using SVD

$$\hat{\mathbf{H}}_\epsilon = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

$$\hat{\mathbf{A}}_a = \mathbf{U}^\top \hat{\mathbf{H}}_a \mathbf{V}\mathbf{D}^{-1}$$

Controlling Overfitting with Hankel Matrices

- To prevent overfitting, control number of states of resulting WFA by

$$\hat{\mathbf{H}} = \underset{\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}}{\text{argmin}} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{H}(x^i), y^i) \quad \text{s.t. } \mathbf{H} \in \text{Hankel}, \ \text{rank}(\mathbf{H}) \leqslant n$$

- Since this is not convex, a usual surrogate is to use Schatten norms

$$\hat{\mathbf{H}} = \underset{\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}}{\text{argmin}} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{H}(x^i), y^i) \quad \text{s.t. } \mathbf{H} \in \text{Hankel}, \ \|\mathbf{H}\|_{S,p} \leqslant R$$

  where $\|\mathbf{H}\|_{S,p} = \|(\mathfrak{s}_1, \ldots, \mathfrak{s}_n)\|_p$ and $\mathfrak{s}_1 \geqslant \cdots \mathfrak{s}_n > 0$ are the singular values of $\mathbf{H}$
- These norms can be computed in polynomial time even for *infinite* Hankel matrices [Balle et al., 2015]

Rademacher Complexity of Hankel Matrices

Given $R > 0$ and $p \geqslant 1$ define the class of infinite Hankel matrices

$$\mathcal{H}_p = \left\{ \mathbf{H} \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star} \;\middle|\; \mathbf{H} \in \text{Hankel}, \|\mathbf{H}\|_{S,p} \leqslant R \right\}$$

Theorem [Balle and Mohri, 2015b]

The Rademacher complexity of $\mathcal{H}_2$ is bounded by

$$\mathfrak{R}_m(\mathcal{H}_2) = O\left(\frac{R}{\sqrt{m}}\right) \quad .$$

The Rademacher complexity of $\mathcal{H}_1$ is bounded by

$$\mathfrak{R}_m(\mathcal{H}_1) = O\left(\frac{R\log(m)\sqrt{W_m}}{m}\right) \quad ,$$

where $W_m = \mathbb{E}_S\left[\min_{\text{split}(S)} \max\left\{\max_p \sum_i 1[p^i = p], \max_s \sum_i 1[s^i = s]\right\}\right]$.

Note: split(S) contains all possible prefix-suffix splits $x^i = p^i s^i$ of all strings in S

Constrained vs. Regularized Optimization

- Constrained ERM with parameter $R > 0$

$$\min_{\mathbf{H}\in\mathbb{R}^{\mathcal{P}\times\mathcal{S}}} \frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{H}(x^i), y^i) \quad \text{s.t. } \mathbf{H} \in \text{Hankel}, \ \|\mathbf{H}\|_{S,p} \leqslant R$$

- Regularized ERM with parameter $\lambda > 0$

$$\min_{\mathbf{H}\in\mathbb{R}^{\mathcal{P}\times\mathcal{S}}} \frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{H}(x^i), y^i) + \lambda\|\mathbf{H}\|_{S,p} \quad \text{s.t. } \mathbf{H} \in \text{Hankel}$$

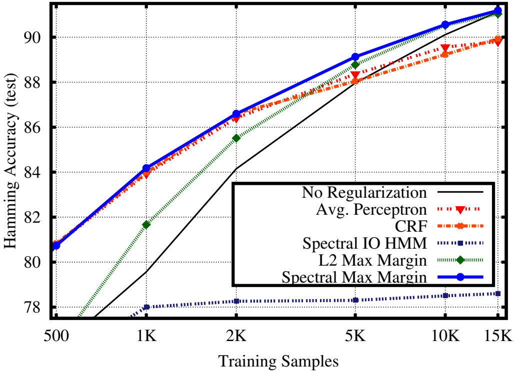
- Regularized versions can be easier to solve and λ easier to tune
- For example, for $\mathcal{H}_2$ bounds *informally* say that for any $\mathbf{H}$

$$L_D(\mathbf{H};\ell) \leqslant \hat{L}_S(\mathbf{H};\ell) + O\left(\frac{\|\mathbf{H}\|_{S,2}}{\sqrt{m}}\right)$$

so choosing $\lambda = O(1/\sqrt{m})$ would imply ERM minimizes a direct upper bound on L_D

Applications of Learning with Hankel Matrices

- Max-margin taggers [Quattoni et al., 2014]



- Unsupervised transducers [Bailly et al., 2013b]
- Unsupervised WCFG [Bailly et al., 2013a]

Conclusion / Open Problems / Future Work

- It is possible to solve *regular inference* with machine learning, focusing on the realistic statistical learning scenario, and still obtain meaningful theoretical guarantees
- In practice works very well, but convex algorithms are not always scalable: we need good implementations
- How to choose $\mathcal{P}$ and $\mathcal{S}$ from data in practice?
- PAC learning of WFA for regression is still open
- Theoretical link between finite and infinite Hankel matrices is still weak

References I

Angluin, D. (1978).
On the complexity of minimum inference of regular sets.
Information and Control, 39(3):337–350.

Angluin, D. and Chen, D. (2015).
Learning a random dfa from uniform strings and state information.
In *International Conference on Algorithmic Learning Theory*, pages 119–133. Springer.

Bailly, R., Carreras, X., Luque, F., and Quattoni, A. (2013a).
Unsupervised spectral learning of WCFG as low-rank matrix completion.
In *EMNLP*.

Bailly, R., Carreras, X., and Quattoni, A. (2013b).
Unsupervised spectral learning of finite state transducers.
In *NIPS*.

Balle, B. and Mohri, M. (2012).
Spectral learning of general weighted automata via constrained matrix completion.
In *NIPS*.

References II

Balle, B. and Mohri, M. (2015a).
Learning weighted automata.
In *Algebraic Informatics*, pages 1–21. Springer.

Balle, B. and Mohri, M. (2015b).
On the rademacher complexity of weighted automata.
In *Algorithmic Learning Theory*, pages 179–193. Springer.

Balle, B., Panangaden, P., and Precup, D. (2015).
A canonical form for weighted automata and applications to approximate minimization.
In *LICS*.

Balle, B., Quattoni, A., and Carreras, X. (2014).
Spectral Learning Techniques for Weighted Automata, Transducers, and Grammars.
http://www.lancaster.ac.uk/~deballep/emnlp14-tutorial/.

References III

Board, R. and Pitt, L. (1992).
On the necessity of occam algorithms.
Theoretical Computer Science, 100(1):157–184.

Chalermsook, P., Laekhanukit, B., and Nanongkai, D. (2014).
Pre-reduction graph products: Hardnesses of properly learning dfas and approximating edp on dags.
In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 444–453. IEEE.

Clark, A. and Thollard, F. (2004).
Partially distribution-free learning of regular languages from positive samples.
In *Proceedings of the 20th international conference on Computational Linguistics*, page 85. Association for Computational Linguistics.

Daniely, A., Linial, N., and Shalev-Shwartz, S. (2014).
From average case complexity to improper learning complexity.
In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 441–448. ACM.

References IV

De la Higuera, C. (2010).
Grammatical inference: learning automata and grammars.
Cambridge University Press.

Fliess, M. (1974).
Matrices de Hankel.
Journal de Mathématiques Pures et Appliquées.

Gold, E. M. (1978).
Complexity of automaton identification from given data.
Information and control, 37(3):302–320.

Hanneke, S. (2016).
The optimal sample complexity of pac learning.
Journal of Machine Learning Research, 17(38):1–15.

Haussler, D., Littlestone, N., and Warmuth, M. K. (1994).
Predicting $\{0, 1\}$-functions on randomly drawn points.
Information and Computation, 115(2):248–292.

References V

Ishigami, Y. and Tani, S. (1993).
The vc-dimensions of finite automata with n states.
In *Algorithmic Learning Theory*, pages 328–341. Springer.

Kearns, M. and Valiant, L. (1994).
Cryptographic limitations on learning boolean formulae and finite automata.
Journal of the ACM (JACM), 41(1):67–95.

Lang, K. J. (1992).
Random dfa's can be approximately learned from sparse uniform examples.
In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 45–52. ACM.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012).
Foundations of machine learning.
MIT press.

Oncina, J. and García, P. (1992).
Identifying regular languages in polynomial time.
Advances in Structural and Syntactic Pattern Recognition, 5(99-108):15–20.

References VI

Parekh, R. and Honavar, V. (2001).
Learning dfa from simple examples.
Machine Learning, 44(1-2):9–35.

Pitt, L. and Warmuth, M. K. (1993).
The minimum consistent dfa problem cannot be approximated within any polynomial.
Journal of the ACM (JACM), 40(1):95–142.

Quattoni, A., Balle, B., Carreras, X., and Globerson, A. (2014).
Spectral regularization for max-margin sequence tagging.
In *ICML*.

Vapnik, V. (1982).
Estimation of dependencies based on empirical data.

Theoretical Guarantees for Learning Weighted Automata

Borja Balle

Mathematics & Statistics | Lancaster University

ICGI Keynote — October 2016